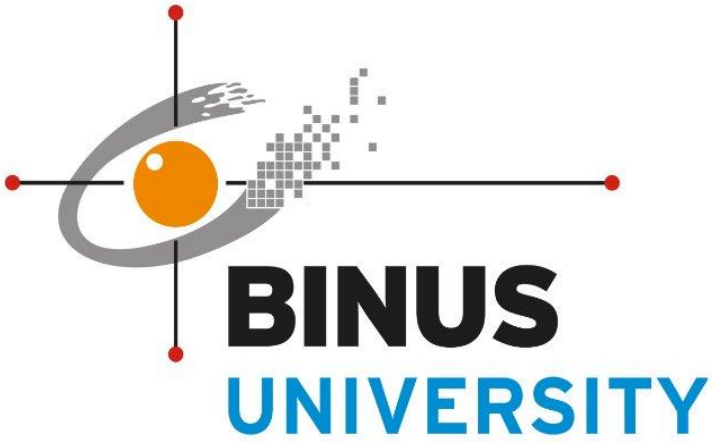


Assignment Cover Letter

(Individual Assignment)

Student Name:

Edsel Putra Harren	2440081942
Course Code : COMP 6699	**Course Name :** Object Oriented Programming
Class : L2BC	**Name of Lecturer :** **Jude Martinez L**
Major : Computer Science	

Title of Assignment :

Vidco-19 Vaccine

Type of Assignment : Final Project

Due Date : 22 - 06 - 2021 **Submission Date : 22 - 06 - 2021**

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer's instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

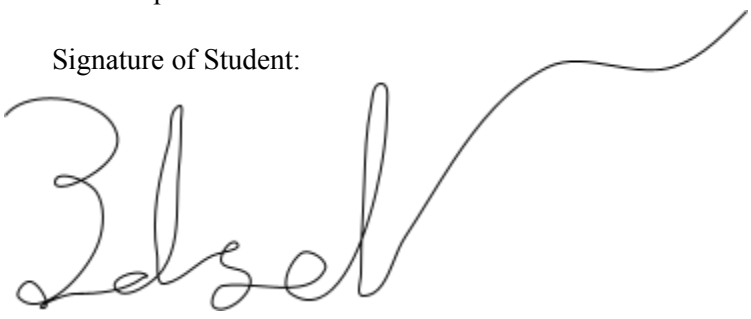
Plagiarism/Cheating

BINUS International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

Declaration of Originality

By signing this assignment, I understand, accept and consent to BINUS International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student: (Name of Student)



Edsel Putra Harren

Table of Contents

Project Specification

The purpose of the program is to make a queue system for people that are going to get a vaccine for the new virus, Covid-19. With the pandemic, everyone is forced to keep a safe distance from each other. The program will help with social distancing because people can enter it through this program and do not need to see people to take the queue for vaccine covid-19 and also have a priority feature for the people who are age 65 above.

More specifications regarding the project are as follows:

- Menu of the program:
    - Option 1 (User data)
        - Name
        - Age
        - Legible or illegible
    - Option 2 (See list of all user)
        - Name,age,legible/illegible, number queue
    - Option 3 (Delete existing user)
        - Need to input the correct name and age
    - Option 4 (Quit and save the input to data.txt)
        - Will sort the data base on age is there are and also will save to data.txt

Solution Design

The main features of the app are:

1. Taking all multiple inputs from the user. (Name,age,legible or illegible)
2. See all the existing user
3. Delete a user, need to input the correct name and age
4. Save the input to Data.txt and also close the program

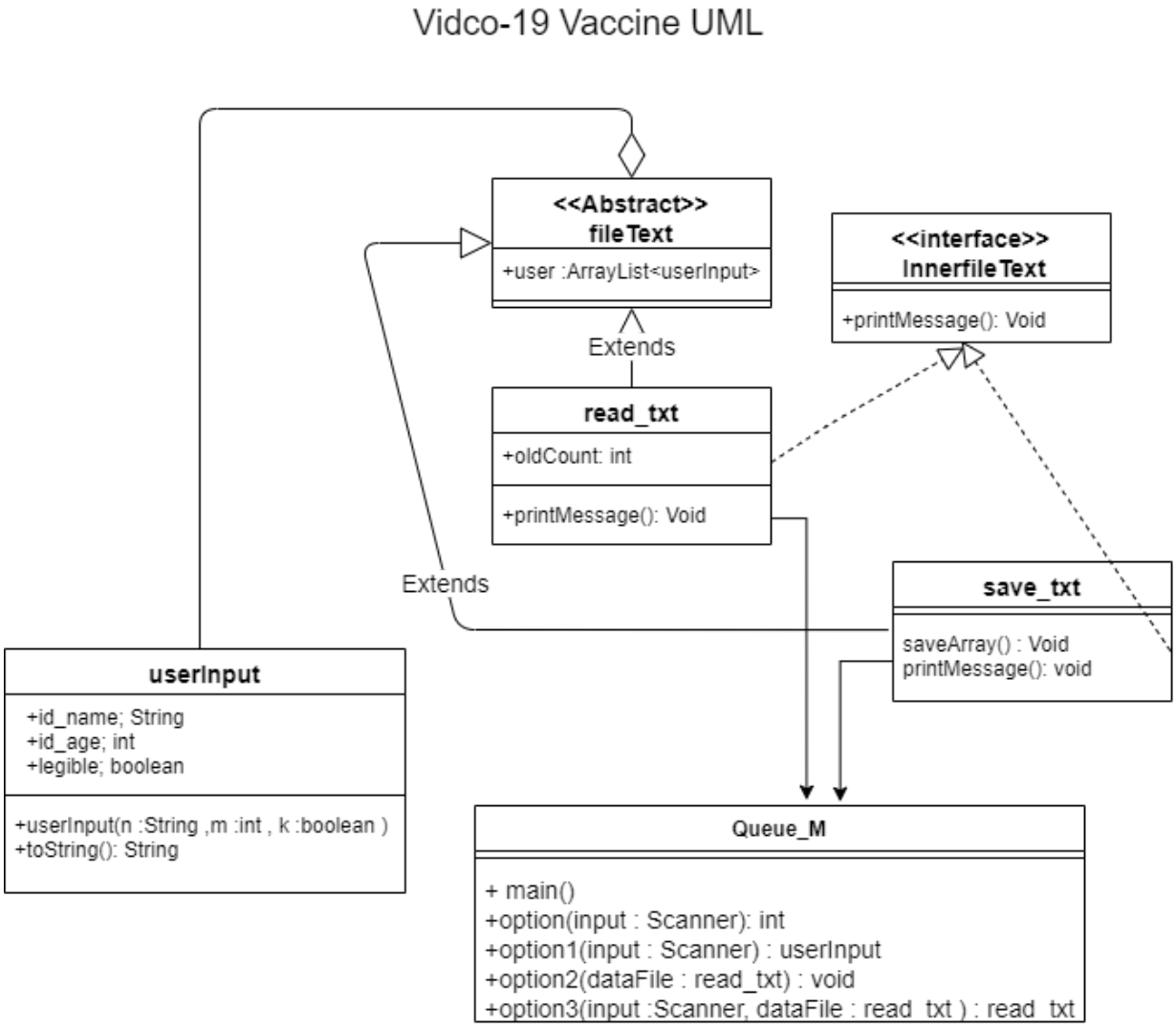
There are 7 files in my program.

- Queue_m.java
- userInput.java
- fileText.java
- innerfileText.java
- Save_txt.java
- Read_txt.java

Implementation

```
import java.util.Scanner; // for input
import java.io.File;// to open file
import java.io.FileNotFoundException;// need for opening file
import java.io.FileWriter;// write the file
import java.io.IOException; // need for write the file
import java.util.ArrayList; /// array list for saving
@Override // to override methods.
read() // read the file
nextLine() // search the next line of the file
size()  // the size of the file
add() // adding the data
sc.close() // closing scan
toString() // change object into a string
```

UML Diagram



How the program works

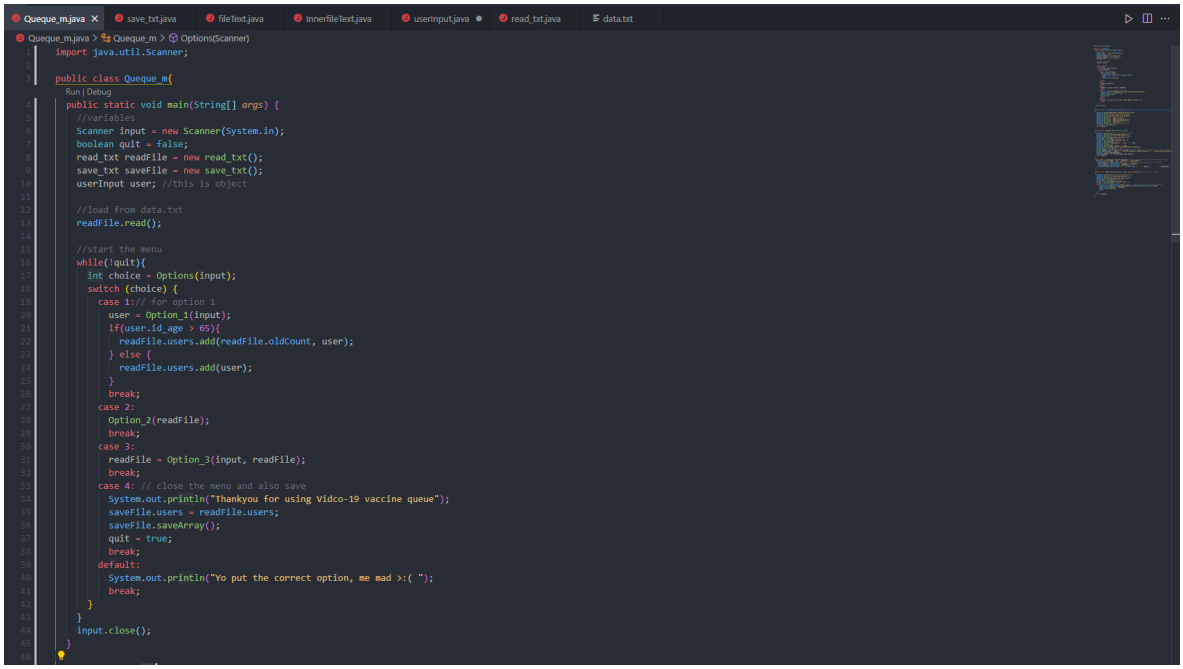
How the saving works and why choose Array list

When I was working on This project, I was thinking that I just need to save the user input to the data.txt file, but when I think again, it was a bad idea for the project to just save an input from the user. Because when the input from the user is just saved to the data.txt, all of the input will be messy and i was thinking on how to make a priority queue for the elderly that are age 65 above. Then I came up with an idea of using an array list. When I think about it again, using an array list to save the input from the user is the best course of action. When using an array list everything will be more organized, also when the program

searches for a specific name it will be easier, and lastly with an array list the program can make a priority queue for the elderly with the age of 65 above. When the program runs the user can input as many as they want. The program will convert every input into an object that will be converted again into an array list. When the program is close the array list will be organize by the program if there's a user that are age 65 above, they will be put in the front

How each file are used

Queue_m.java

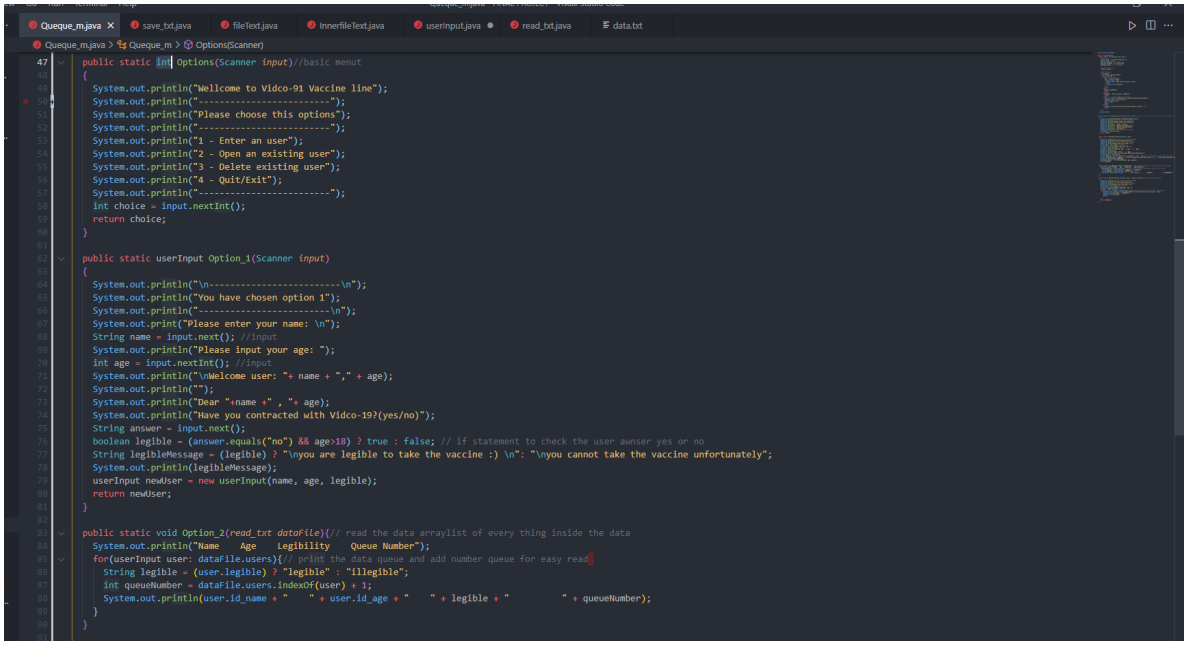


```java
import java.util.Scanner;

public class Queue_m{
    Run | Debug
    public static void main(String[] args) {
        //variables
        Scanner input = new Scanner(System.in);
        boolean quit = false;
        read_txt readFile = new read_txt();
        save_txt saveFile = new save_txt();
        userInput user; //this is object

        //load from data.txt
        readFile.read();

        //start the menu
        while(!quit){
            int choice = Options(input);
            switch (choice) {
                case 1:// for option 1
                    user = Option_1(input);
                    if(user.id_age > 65){
                        readFile.users.add(readFile.oldCount, user);
                    } else {
                        readFile.users.add(user);
                    }
                    break;
                case 2:
                    Option_2(readFile);
                    break;
                case 3:
                    readFile = Option_3(input, readFile);
                    break;
                case 4: // close the menu and also save
                    System.out.println("Thankyou for using Vidco-19 vaccine queue");
                    saveFile.users = readFile.users;
                    saveFile.saveArray();
                    quit = true;
                    break;
                default:
                    System.out.println("Yo put the correct option, me mad >:( ");
                    break;
            }
        }
        input.close();
    }
```



```java
    public static int Options(Scanner input)//basic menut
    {
        System.out.println("Wellcome to Vidco-91 Vaccine line");
        System.out.println("-------------------------");
        System.out.println("Please choose this options");
        System.out.println("-------------------------");
        System.out.println("1 - Enter an user");
        System.out.println("2 - Open an existing user");
        System.out.println("3 - Delete existing user");
        System.out.println("4 - Quit/Exit");
        System.out.println("-------------------------");
        int choice = input.nextInt();
        return choice;
    }

    public static userInput Option_1(Scanner input)
    {
        System.out.println("\n-------------------------\n");
        System.out.println("You have chosen option 1");
        System.out.println("-------------------------\n");
        System.out.print("Please enter your name: \n");
        String name = input.next(); //input
        System.out.println("Please input your age: ");
        int age = input.nextInt(); //input
        System.out.println("\nWelcome user: "+ name + "," + age);
        System.out.println("");
        System.out.println("Dear "+name +" , "+ age);
        System.out.println("Have you contracted with Vidco-19?(yes/no)");
        String answer = input.next();
        boolean legible = (answer.equals("no") && age>18) ? true : false; // if statement to check the user awnser yes or no
        String legibleMessage = (legible) ? "\nyou are legible to take the vaccine :) \n": "\nyou cannot take the vaccine unfortunately";
        System.out.println(legibleMessage);
        userInput newUser = new userInput(name, age, legible);
        return newUser;
    }

    public static void Option_2(read_txt dataFile){// read the data arraylist of every thing inside the data
        System.out.println("Name    Age    Legibility    Queue Number");
        for(userInput user: dataFile.users){// print the data queue and add number queue for easy read
            String legible = (user.legible) ? "legible" : "illegible";
            int queueNumber = dataFile.users.indexOf(user) + 1;
            System.out.println(user.id_name + "     " + user.id_age + "    " + legible + "          " + queueNumber);
        }
    }
```

```
public static read_txt Option_3(Scanner input, read_txt dataFile)//delete the data you want
{
  System.out.println("\n-------------------------\n");
  System.out.println("You have chosen option 3");
  System.out.println("-------------------------\n");
  System.out.print("Please enter the name: \n");
  String name = input.next(); //input
  System.out.println("Please input their age: ");
  int age = input.nextInt(); //input
  for(int i = 0; i < dataFile.users.size(); i++){// for searching each stuff inside the arraylist
    if(dataFile.users.get(i).id_name.equals(name) && dataFile.users.get(i).id_age == age){
      System.out.println(name + " deleted!");
      dataFile.users.remove(i);
      break;
    }
  }
  return dataFile;
 }
}
```

Queue_m.java This file is for the user to start the program, this will also be the file for the menu for the program.

```
public static int Options(Scanner input)//basic menut
{
  System.out.println("Wellcome to Vidco-91 Vaccine line");
  System.out.println("-------------------------");
  System.out.println("Please choose this options");
  System.out.println("-------------------------");
  System.out.println("1 - Enter an user");
  System.out.println("2 - Open an existing user");
  System.out.println("3 - Delete existing user");
  System.out.println("4 - Quit/Exit");
  System.out.println("-------------------------");
  int choice = input.nextInt();
  return choice;
}
```

is the main code to run the program, and also where the menu is.

Then I also use switch choice to make the while loop to be clean looking

```java
import java.util.Scanner;

public class Queque_m{
  Run | Debug
  public static void main(String[] args) {
    //variables
    Scanner input = new Scanner(System.in);
    boolean quit = false;
    read_txt readFile = new read_txt();
    save_txt saveFile = new save_txt();
    userInput user; //this is object

    //load from data.txt
    readFile.read();

    //start the menu
    while(!quit){
      int choice = Options(input);
      switch (choice) {
        case 1:// for option 1
          user = Option_1(input);
          if(user.id_age > 65){
            readFile.users.add(readFile.oldCount, user);
          } else {
            readFile.users.add(user);
          }
          break;
        case 2:
          Option_2(readFile);
          break;
        case 3:
          readFile = Option_3(input, readFile);
          break;
        case 4: // close the menu and also save
          System.out.println("Thankyou for using Vidco-19 vaccine queue");
          saveFile.users = readFile.users;
          saveFile.saveArray();
          quit = true;
          break;
        default:
          System.out.println("Yo put the correct option, me mad >:( ");
          break;
      }
    }
    input.close();
  }

```

option 1 they can input their name ,age, and there will be a question that will determine if the user is eligible to take the queue or not. Because of the input of the user, the program needs to save the input. So the program will convert the user input into an object type, and then will be converted into an Array list. With an Array list, the user input will be more organized and also much easier to search for a specific name by the program.

```java
public static userInput Option_1(Scanner input)
{
  System.out.println("\n-------------------------\n");
  System.out.println("You have chosen option 1");
  System.out.println("-------------------------\n");
  System.out.print("Please enter your name: \n");
  String name = input.next(); //input
  System.out.println("Please input your age: ");
  int age = input.nextInt(); //input
  System.out.println("\nWelcome user: "+ name + "," + age);
  System.out.println("");
  System.out.println("Dear "+name +" , "+ age);
  System.out.println("Have you contracted with Vidco-19?(yes/no)");
  String answer = input.next();
  boolean legible = (answer.equals("no") && age>18) ? true : false; // if statement to check the user awnser yes or no
  String legibleMessage = (legible) ? "\nyou are legible to take the vaccine :) \n": "\nyou cannot take the vaccine unfortunately";
  System.out.println(legibleMessage);
  userInput newUser = new userInput(name, age, legible);
  return newUser;
}
```

Option 2 will check all of the inside of the data.txt and then will print the data in the terminal with a range of name,age,legible/illegible, queue number. This option is useful if the operator/user wants to see their data.

```java
public static void Option_2(read_txt dataFile){// read the data arraylist of every thing inside the data
  System.out.println("Name    Age    Legibility    Queue Number");
  for(userInput user: dataFile.users){// print the data queue and add number queue for easy read
    String legible = (user.legible) ? "legible" : "illegible";
    int queueNumber = dataFile.users.indexOf(user) + 1;
    System.out.println(user.id_name + "    " + user.id_age + "    " + legible + "          " + queueNumber);
  }
}
```

Option 3 is to delete a user, this is useful if the user wants to delete their data. To use option 3 the user needs to input a specific name and also age.

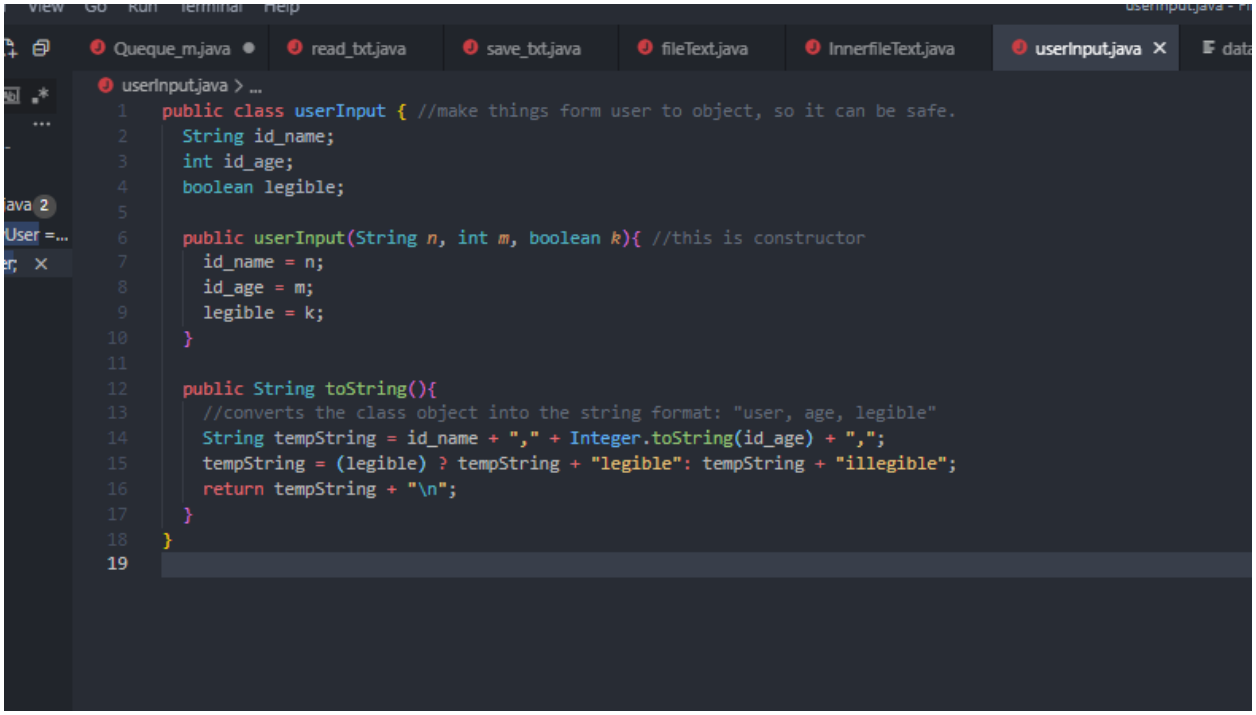
```java
}

public static read_txt Option_3(Scanner input, read_txt dataFile)//delete the data you want
{
  System.out.println("\n-------------------------\n");
  System.out.println("You have chosen option 3");
  System.out.println("-------------------------\n");
  System.out.print("Please enter the name: \n");
  String name = input.next(); //input
  System.out.println("Please input their age: ");
  int age = input.nextInt(); //input
  for(int i = 0; i < dataFile.users.size(); i++){// for searching each stuff inside the arraylist
    if(dataFile.users.get(i).id_name.equals(name) && dataFile.users.get(i).id_age == age){
      System.out.println(name + " deleted!");
      dataFile.users.remove(i);
      break;
    }
  }
  return dataFile;
}
}
```

Option 4 is to save the input of the user to the data.txt and also to end the program.

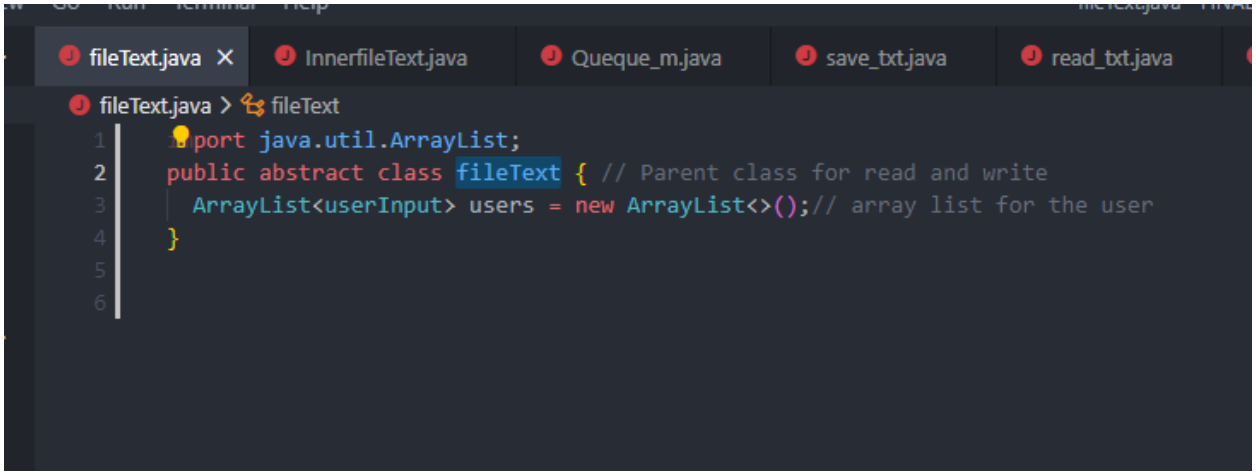
```java
        case 4: // close the menu and also save
          System.out.println("Thankyou for using Vidco-19 vaccine queue");
          saveFile.users = readFile.users;
          saveFile.saveArray();
          quit = true;
          break;
        default:
          System.out.println("Yo put the correct option, me mad >:( ");
          break;
      }
    }
    input.close();
```

userInput.java



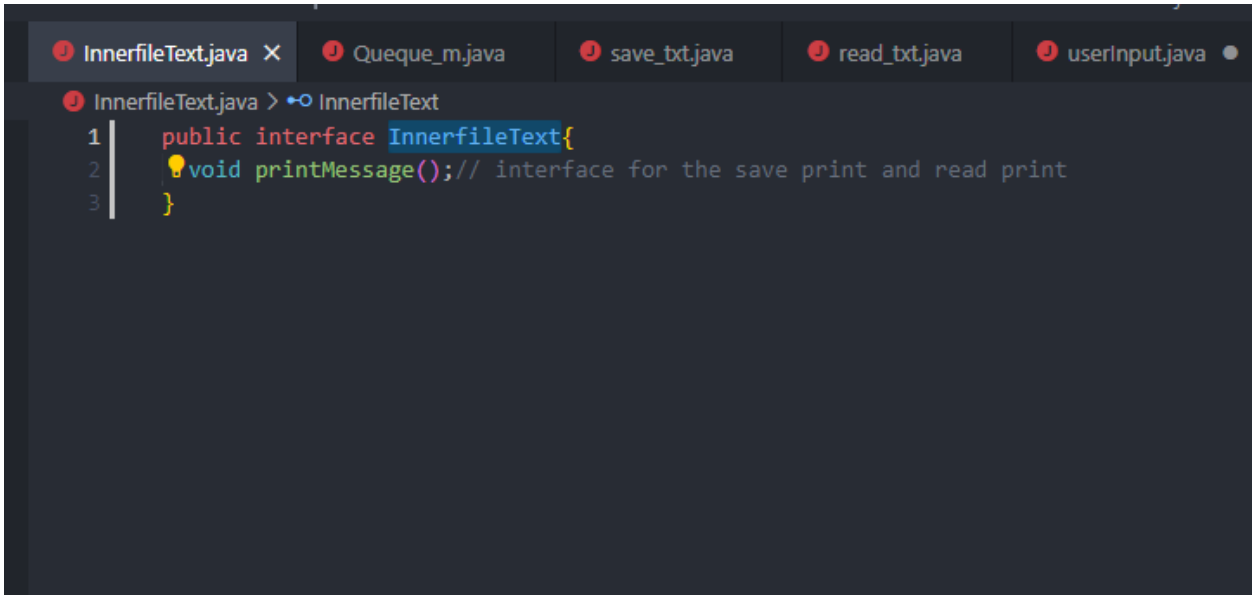
to change the type of the input from string,int and bool, into an object.The userInput() is for the constructor and the toString() is for to change the format into an string.

fileText



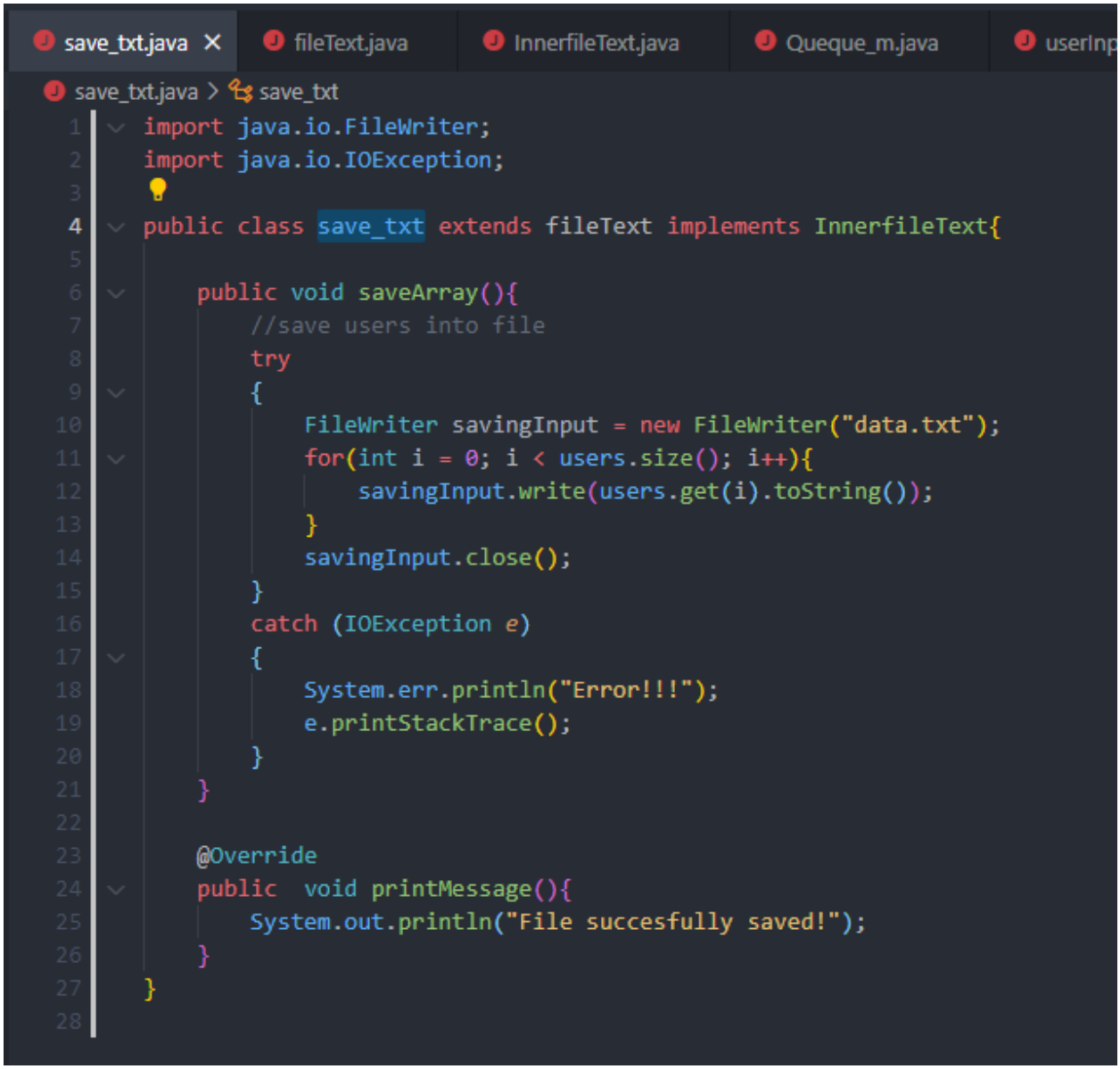
This class is to make the array list, and also make the variable called users.

InnerfileText



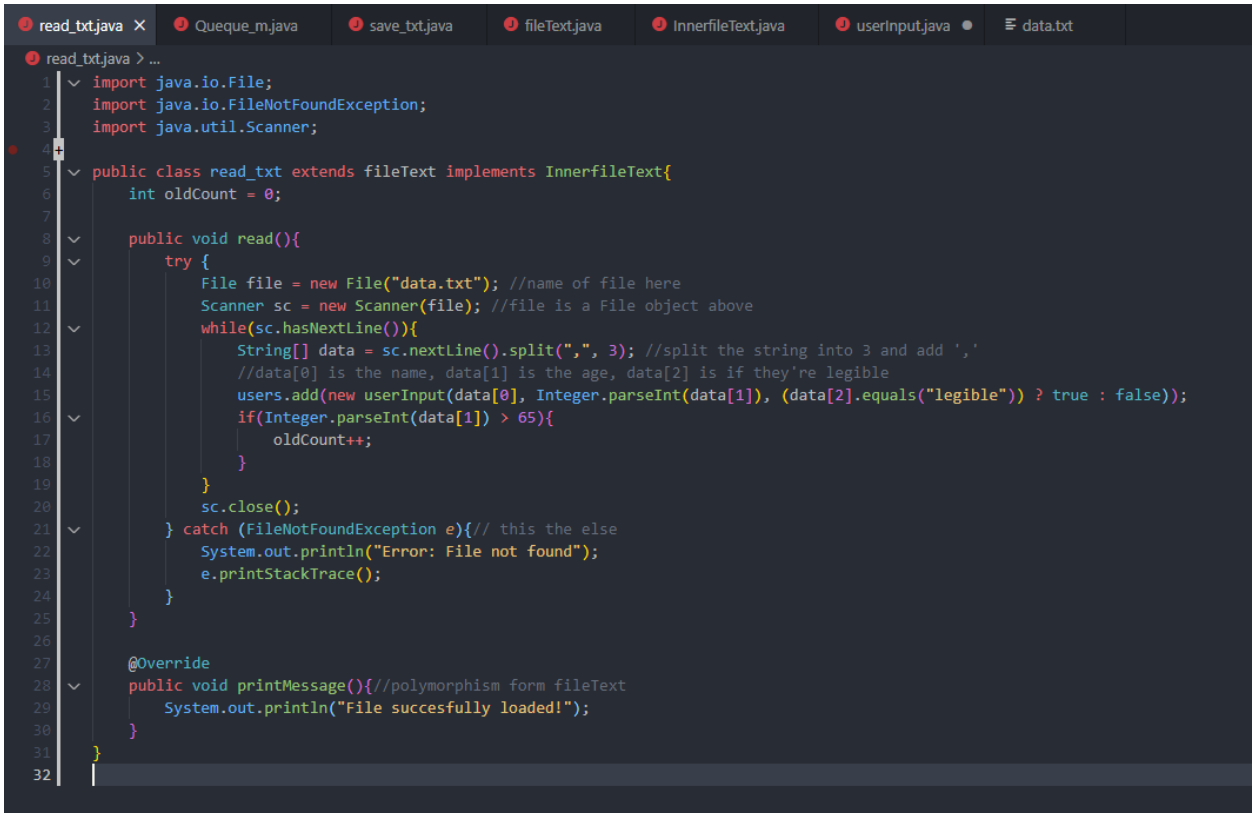
The printMessage() is going to be used in the **save_txt** and also the **read_txt**, this method is for printing a specific message in the **save_txt** and **read_txt**.

save_txt



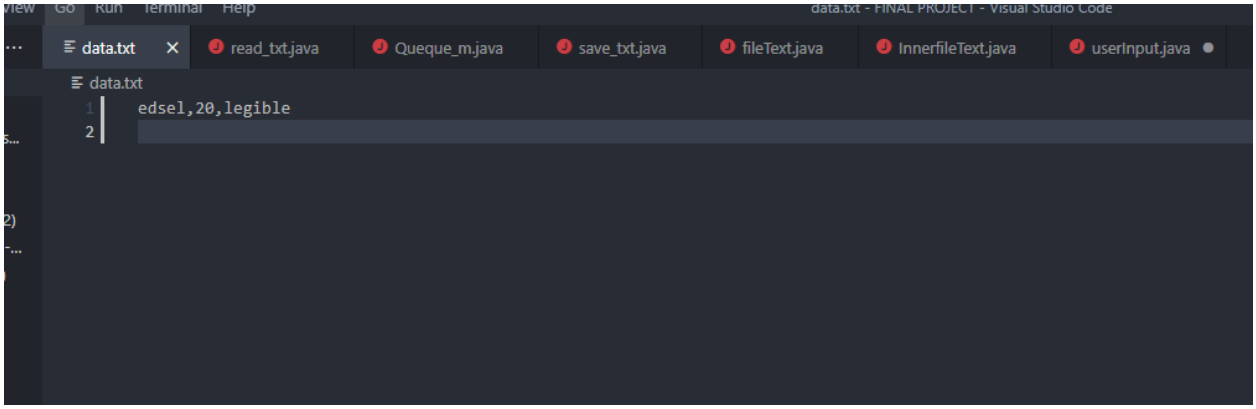
This file is to save the input from the user. The save_txt.java needs to extend the **fileText.java** for the array list, and needs to implement the **InnerfileText.java** to print if the file is successful to save or not. The program will put the input from user into the **userInput.Java,** after the data being converted into an object and then into a string, it's going to be save by this class. By using saveArray() it will use the filewriter to save the input into a data.txt file. The if statement is to keep checking the input and save it into the **data.txt**, it will keep checking the input till there's no more input from the user.

read_txt



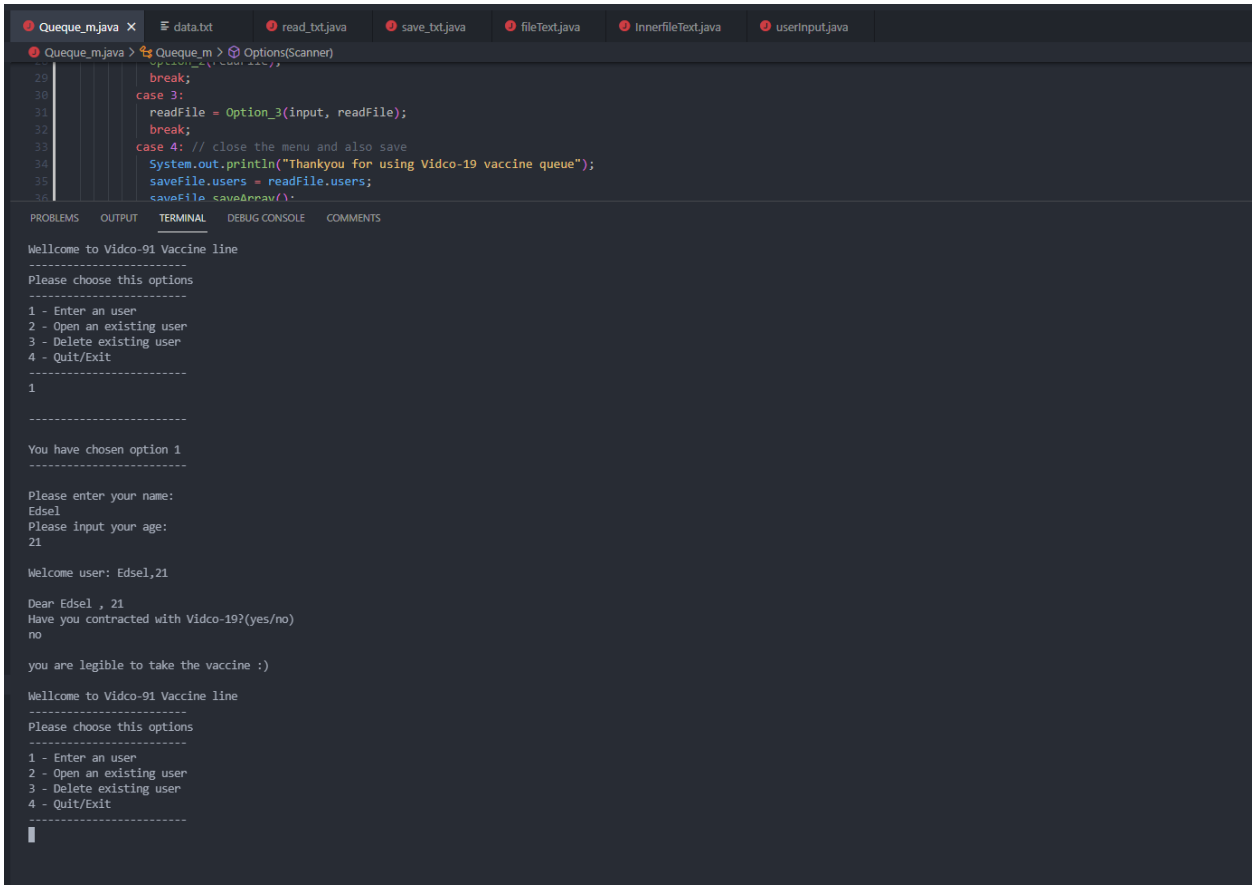
The **read_txt.java** will extend from the **fileText.java** to get the array and then it will implement the **InnerfileText.java** to print if the file is loaded or not . This is to be used for reading the input from the user and also the data.txt, then it will print the data into the terminal. **read()** is for the program to read all of the data in the input then it will combine it with the user input. It will keep adding new user, until the user end the program where the data will be stored in the **data.txt** . also the oldCount is the priority queued, the program will check if the age of each user, then if the program finds that there's a user that are above the age of 65 it will push the user into the index 0, then if there's also another old person it will keep adding after the first old user. That's the oldCount ++ for.

data.txt

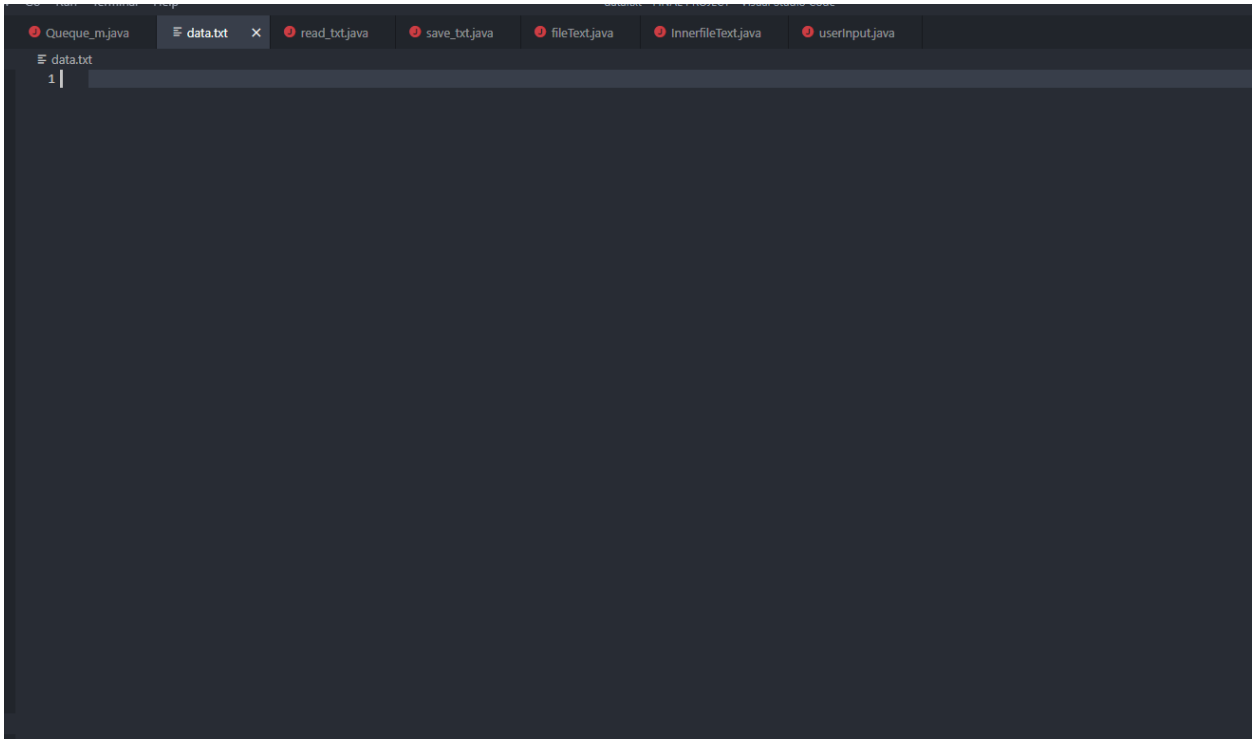


This is where the input from the user will be saved, the data.txt will have the following format to save name,age,then legible/illegible.

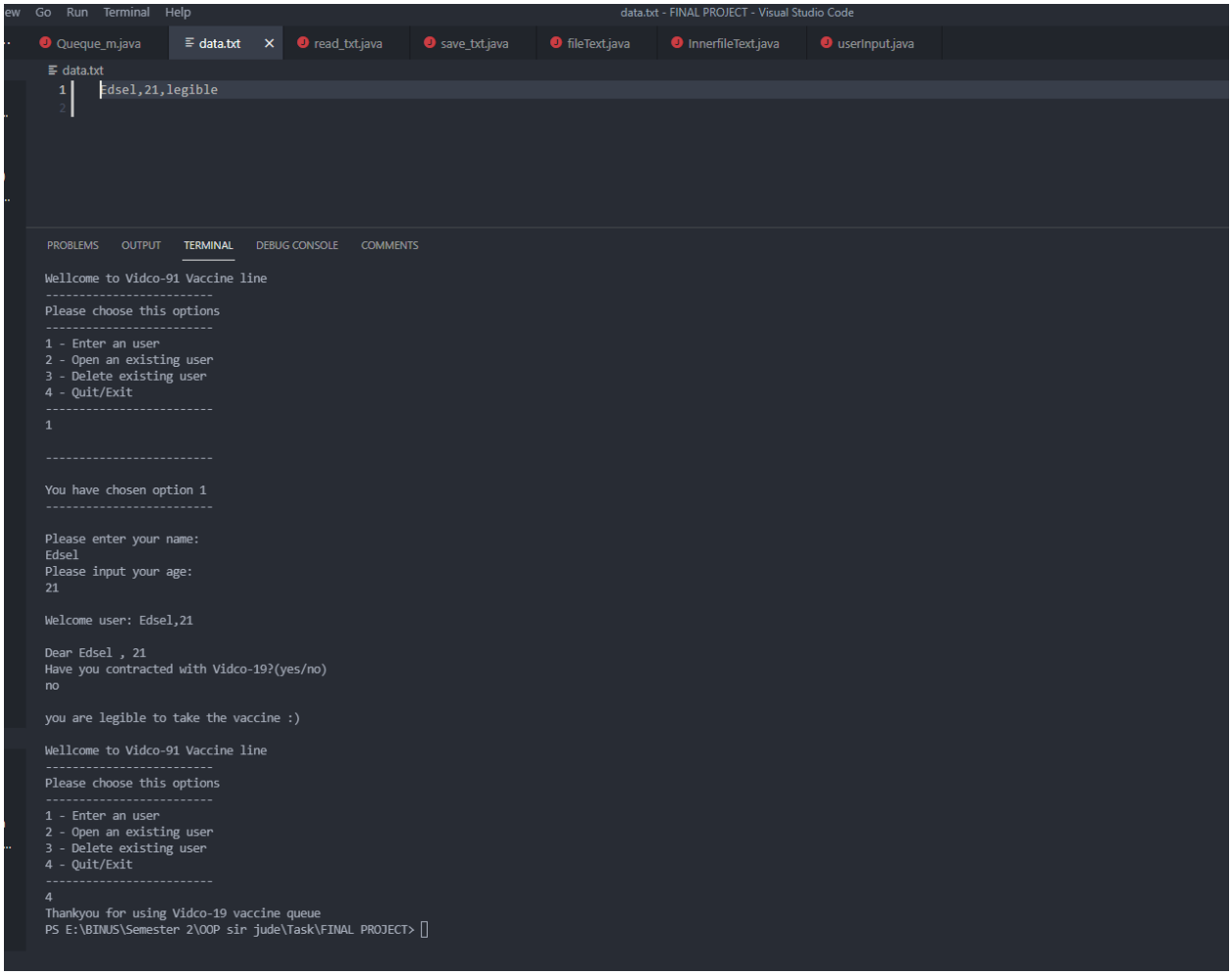
Screenshots of Working Program



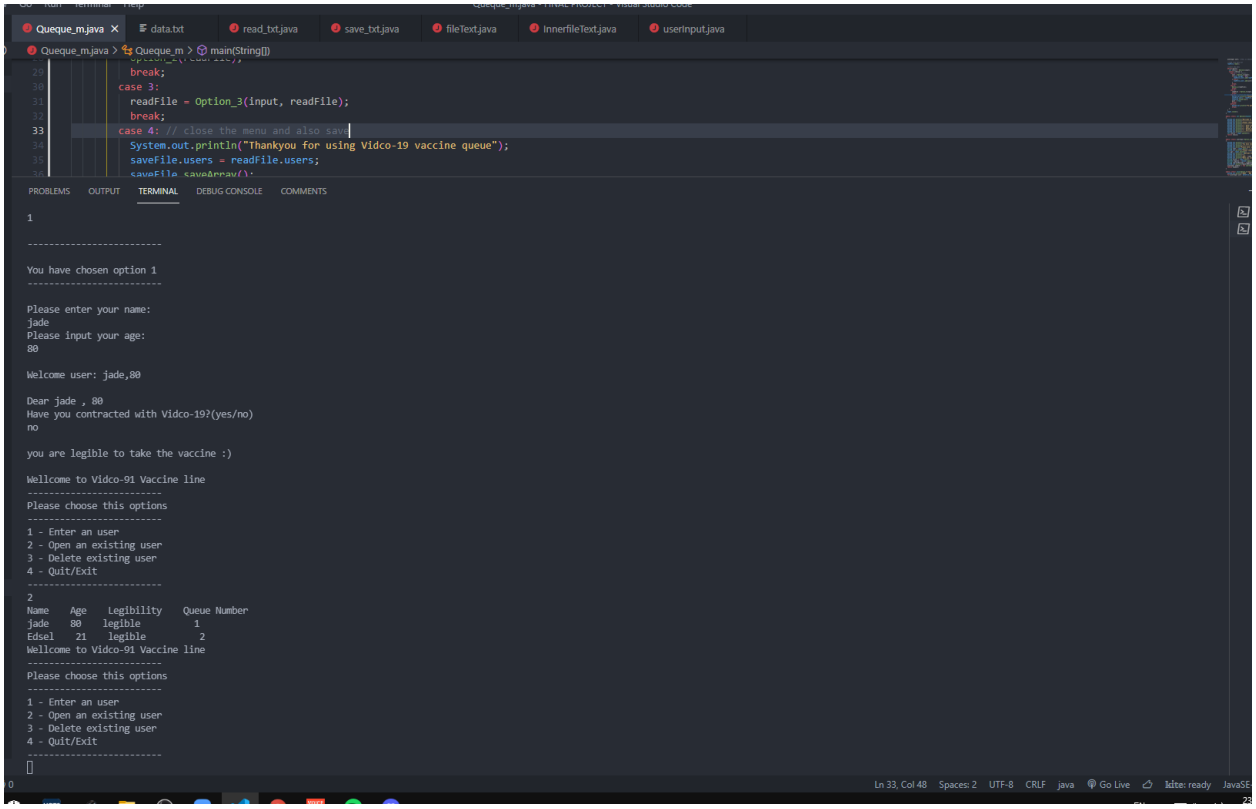
This is the program running



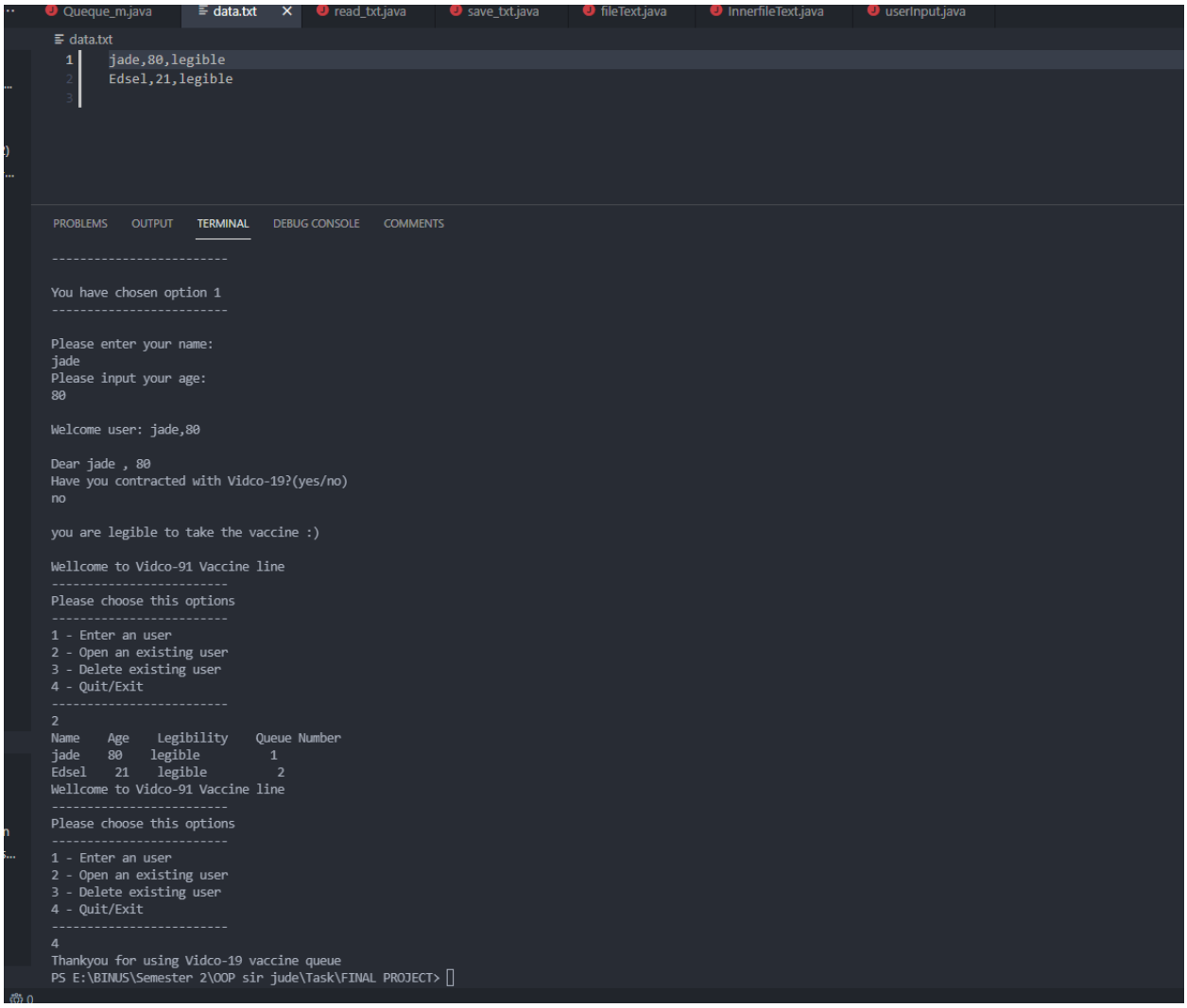
Empty data. This is the proof that the data is still empty and have not been added.



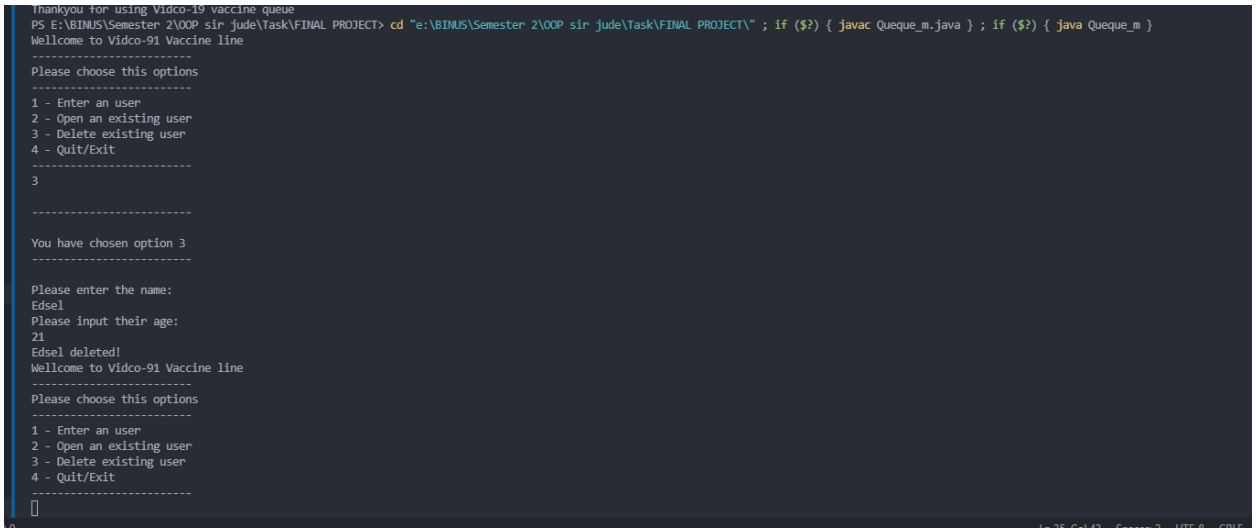
This is after the program is save and quit.



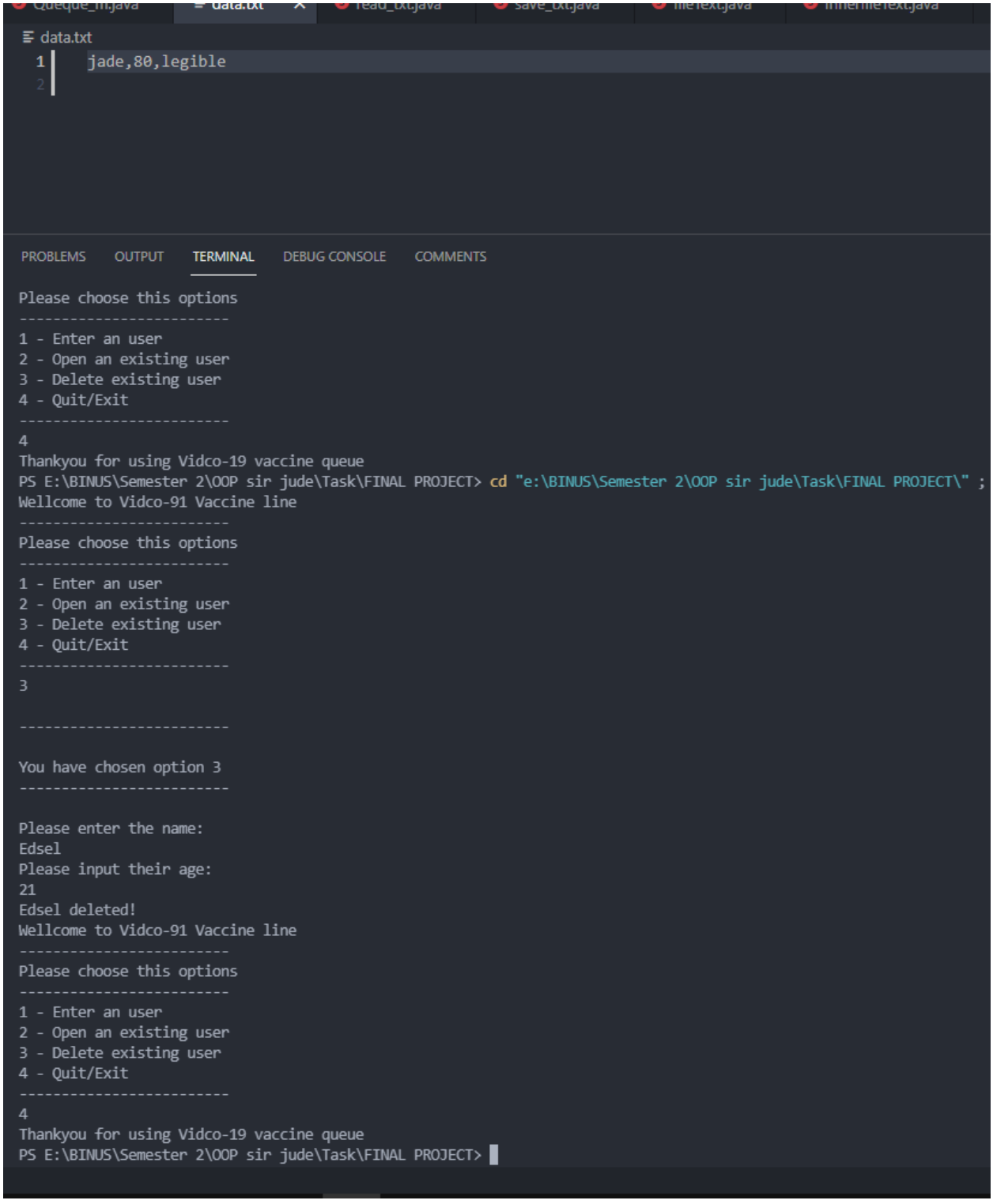
this is adding a new user name jade who is 80 years old, and this is the program if the user choose option 2, it will show all the user.



Jade is being put into the index 0 because Jade is in the priority queue because of the age.



This screen shot is to show the program successfully delete the user.



after the user choose option 4, save and close, my data will be updated.